StephAmon

Ambulomancy:

March 2005

Predicting Roaming Walks in Pharaoh

Pharaoh's walkers can drive you mad. You know they can make long walks, servicing a huge housing block or keeping a massive industrial block working smoothly, but the little devils persist in taking a lot of useless short walks or heading resolutely in the worst possible direction. If you have had quite enough of this nonsense, then read on. Roaming walker behavior can not only be predicted, it can also be manipulated to send those miscreants exactly where you want them to go.

Best of all, no great mathematical expertise or insight into computer program architecture is needed to tame the roamers. Once you know the tricks and rules, all you have to do to use them is to count squares, and Pharaoh players tend to have become better at square counting than most of us would have preferred. However, you may need to do a little preliminary reading before going any further in this discussion. If you are not already familiar with it, please read Brugle's invaluable "Random walker start and finish points" in the Game Help Forum on PharaohHeaven. This was a ground-breaking article in roaming-walker research and has saved the bacon of many a Pharaoh player trying to figure out why a particularly idiotic roamer was failing to loop or how in blazes the magistrate escaped from a housing block. The text below assumes knowledge of the walk-start squares described in that document.

Acknowledgments. I am most grateful to Brugle for his critical review of an earlier draft of this paper, which weeded out a good many ambiguities, inconsistencies and at least one out-right error. If I managed to sneak any such flaws back into the article during its revision, that is my fault. Brugle did his best to save me.

Walk Modes and Durations

Destination mode. The game manual divides walkers into two types, roaming and destination, but we know that even destination walkers make more than one kind of walk. Storage yard cartmen on a "get" go cross-country ignoring roads and using ferries as needed. Military companies travel off road but will not use ferries. However, the majority of destination walkers (like farm and industry delivery men, granary cart pushers, and bazaar buyers) must travel on roads, may pass through roadblocks, and may use ferries. All roaming walkers also enter a similar kind of destination mode both during the out-bound half of many of their walks (as we will see below) and when returning to their buildings. In this kind of destination walk, the roamer has a fixed objective and takes the shortest available path to it subject to the rules that he must stick to roads, may not use ferries, but may pass through roadblocks. This is the kind of walk that will be said to be in ***destination mode*** in this document.

An individual walk by a roaming walker switches modes. Roamers leave their buildings in a walk mode (or modes) that certainly looks rather random and hard to predict. Here, this will be called the roaming half of the roamer's walk. Predicting the roaming half of a roamer's walk is the subject of this paper. At some point along his walk, the roamer reaches a go-home point and switches to (road-bound, roadblock-violating) destination mode for the return trip to his building.

Roaming walks and random mode. A small number of the walks made by the roaming walkers in a typical Pharaoh city have a roaming half of the walk that is almost entirely in a single mode that will be called ***random mode*** here. Roamers walking in random mode are free to turn in different directions unpredictably at intersections, but may not pass through roadblocks (or use ferries or leave the road). However, roamers appear capable of making four different kinds of walks for the roaming half of an individual walk, only one of which is in nearly pure random mode. As we will see below, one of the roaming walks is made in pure destination mode and the other two are hybrids, beginning in destination mode and then switching to random mode until the go-home point is reached.

Roamer endurance classes. Some roamers simply travel further than others. All roamers can be grouped into three classes based on the lengths of their shortest walks. Most roamers are ***short walkers***. The shortest walk they ever make before conversion to home-bound destination mode is 26 squares, if you begin by counting the walk-start square itself as square zero. The game's ***medium walkers*** are dancers, jugglers, musicians, senet players, tax collectors, and zoo keepers. Their shortest walks last 35 squares to the go-home point. ***Long walkers*** include the architect, fireman, policeman, and magistrate. On their shortest walks, they travel 43 squares before switching to destination mode for the journey home. The text below requires frequent reference to the length of the shortest (default) walk for a walker of a given endurance class, so it will be hereafter symbolized as ***d***.

The Walk Cycle

Routing centers. Each walk executed by a roamer is formally associated with a semicardinal compass direction, like northeast or southwest in contrast to a cardinal direction like due north. During the generation of each roaming walk, the algorithm bases its decision about which of the four kinds of roaming walks the roamer will follow (and what its initial path or direction will be) upon the pattern of roads that it finds around a ***routing center*** located eight map squares from the north square of the roamer's building in the semicardinal direction formally associated with the walk. The largest area that the algorithm is prepared to examine around a routing center is a 13x13 square area (with the routing center in the very middle) called a ***routing zone***. Usually, however, the algorithm does not need to search an entire routing zone to determine the kind of roaming walk to issue for the roamer associated with that zone.

Since there are four semicardinal directions, each roamer's building has four routing centers eight squares away from the building's north square in each of these four directions. On successive walks, the algorithm consults each of the building's routing centers one after the other in clockwise order. When a roamer-dispatching building is initially placed on the map, we cannot know in advance which semicardinal direction will be the first to be examined by the algorithm, but if the first walk is associated with the northeast routing center, the next walk will be associated with the southeast routing center, which is the next one in clockwise rotation.

Quadrambles. A building has only four different routing centers, so the algorithm can only generate four completely different walks for the building's roamer before it is forced to start reusing the routing centers (in the same clockwise order it checked them for the first four walks). Roaming walks are, thus, generated in cycles of four walks, in which one walk per cycle is associated with each of the four semicardinal directions and with the routing center lying in that direction from the building. In postings to Pharaoh Heaven, the author coined the term ***quadrambles*** (Latin for "four walks") for these walk cycles. Each of the four walks in a quadramble is called a ***leg*** of the quadramble. The remainder of this work is devoted to predicting individual quadramble legs, but it is important to remember that roaming walkers execute these walks as part of a four-walk cycle.

Walk-Type Selection

To generate a roaming walk, the algorithm searches for road squares around the associated routing center in a predictable order. The first 49 squares searched within a routing zone are shown in Table 1. Conceptually, it is helpful to regard the squares of the routing zone as being organized into concentric square ***search rings***. Three search rings are shown using different colors in Table 1.

Table 1. The first 49 squares around a routing center that the algorithm checks for roads numbered in the order in which they are examined. The routing center (white cell at center) is the fifth square checked for a road. North lies toward upper left.

26	27	28	29	30	31	32
33	10	11	12	13	14	34
35	15	1	2	3	16	36
37	17	4	5	6	18	38
39	19	7	8	9	20	40
41	21	22	23	24	25	42
43	44	45	46	47	48	49

The search order shown in the table is strikingly similar to the order in which the game searches for walk-stop squares (as described by Brugle in “Random walker start and finish points”. The innermost (yellow in Table 1) search ring is examined in the same order that the game uses for seeking walk-finish squares around a 1x1 building, and the second (green) ring is searched in the same order in which walk-finish squares are found around a 3x3 building. Walk-finish squares for the labor recruiter from a storage yard and for a priest from a temple complex are prioritized in the same order as map squares within two concentric routing search rings, although they are necessarily incomplete where the buildings get in the way. This same, left-to-right, top-to-bottom pattern is followed in searching the rest of the 13x13 routing zone, unless road discovery terminates the search.

Absent and disconnected roads. If no roads at all are found during the search of a routing zone, then a default walk (see below) is generated for the quadramble leg corresponding to that zone. Disconnected roads in a routing zone can also lead to the generation of a default walk. A disconnected road square is one from which no completely road-connected path can be traced to the walk-start square of a walker’s building. The length of such a road-connected path is irrelevant to the identification of a road as connected or disconnected, as is the presence or absence of roadblocks along that connecting road. If a search of a routing zone finds only disconnected road squares, then a default walk is generated.

When both connected and disconnected road squares occur in a routing zone, then a default walk is issued if the connected road square closest to the routing center lies in a search ring outside the search ring containing the disconnected road square nearest to the search center. In addition, the algorithm’s search of the routing zone ends after examining the last square in the search ring containing the first-found disconnected road square. Until the search of that ring is complete, however, a connected road square may still be discovered.

Default walks. A default walk has length equal to ***d*** and is entirely in random mode except between the walk-start square and the first square of travel. If the walk-start square falls on a corner or straight road, initial direction is determined by a simple series of preferences: NE > SE > NW > SW. If the walk-start square falls in a straight piece of road, the roamer’s default initial directional preference on a straight road can be summarized succinctly as “East”. The addenda provide notes on problems with the rare case of default walks from three-way intersections. After the first square of travel on a default walk, the roamer is in pure random mode. He does not reverse directions unless a dead end or roadblock forces him to, but otherwise he is free to turn unpredictably at intersections.

Connected roads. When the algorithm's search of a routing zone encounters a connected road square, the search ends immediately, and that road square becomes the ***walk target***. The algorithm also identifies the shortest road-connected path, called the ***shortest circuit***, leading from the building's walk-start square to the walk target. If two different connecting paths have equal length and shorter length than all others between walk-start and target, then the shortest circuit is selected as described in the section on Shortest Circuit Tie-breaking in the Addenda below.

Shorting due to blockage. When a walk target has been identified, the algorithm searches the shortest circuit for roadblocks from the walk-start square either to the walk-target or to the ***d***-th square, whichever distance is shorter. If no roadblock is found, then the length of the shortest circuit is evaluated as described in the next section. If a roadblock is found, then shorting due to blockage occurs.

A blockage-shorted walk is a hybrid-mode walk with outbound length equal to ***d***. It begins with a destination walk from the walk-start square to the last square along the shortest circuit before the roadblock (or ferry landing, as described below). On that last walkable square, a conversion occurs to random mode for an additional number of squares of travel exactly sufficient to make the total length of the roaming half of the walk equal to ***d***. The last square of the random portion of this walk is the go-home point at which the roamer reverts to destination mode for the return to his building. During the destination mode portion of the roaming half of this walk, the walker is bound to the shortest path and is not free to make unpredictable turns into stub roads and the like.

At the outbound mode-conversion point, the roamer does not reverse direction. If the conversion occurs in the middle of a straight piece of road, the roamer just keeps going in the same direction in which he had been traveling earlier in destination mode. If the conversion occurs in the middle of an intersection, the roamer may choose any of the available roads except the one via which he arrived at the intersection.

Shorting due to distance. When a blockage-free walk-target has been identified, the algorithm compares the length of the shortest circuit to ***d***. If shortest-circuit length is strictly less than ***d*** (not "less than or equal" to ***d***), then the target is deemed to be walkable, no shorting occurs, and a grounded walk (described below) is issued. If shortest circuit length equals or exceeds ***d***, then shorting due to distance occurs. A distance- shorted roaming walk has length ***d***, and remains entirely in destination-mode all the way to the go-home point. The algorithm sends the roamer on a destination walk along the shortest circuit to the ***d***-th square, which serves as the go-home point. No unpredictable departures from the shortest circuit are permitted during this kind of roaming walk.

Grounded walks. When a walkable target has been identified, the algorithm generates a grounded walk. A grounded walk has variable length ranging up to 2***d***-1. A grounded walk is a hybrid beginning with a destination walk from the walk-start square to the walk target. Mode conversion occurs on the walk target square itself, and the roamer continues for an additional ***d*** squares of travel in random mode. Again, the roamer is initially bound to the shortest circuit, from which no deviations are permitted, until the target square is reached. At the point of conversion to random mode on the target, the roamer does not reverse direction but otherwise is free to choose any other available direction, if the conversion occurs in the middle of an intersection.

Doubly shorted walks. A few walkers can become "confused" if they are simultaneously shorted both by blockage and by distance, i.e., if the walk target lies *d* squares or further from the walk start square along the shortest circuit *and* a road block a road block appears within the first *d* - 1 squares along that circuit. A confused walker may leave the road entirely. Only certain roamers become confused. Confused walks are discussed at the end of this monograph in the section titled "Walks confused by blockage and distance".

Illustrations

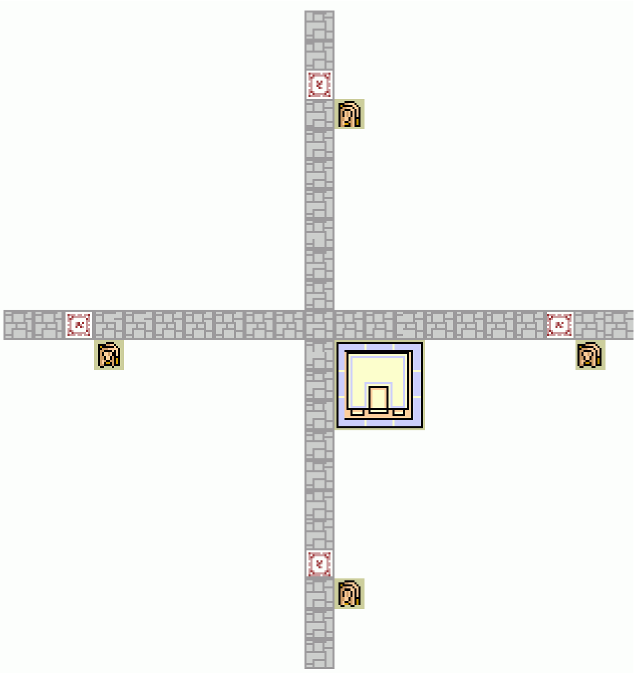

Figure 1. The four routing centers (small statues) eight squares in all four semicardinal directions from the north square of a temple on a four-way intersection. North must lie toward upper left or the temple's north square and, therefore, the routing centers will shift.

In Fig. 1 the four routing centers of a temple are marked by small statues. When the algorithm begins searching the routing zones around these centers, in every case a connected road square is discovered in the first location examined (the one due north of the routing center). These first-found connected road squares are denoted by plaza tiles in the figure, and each one is located eight squares from the four-way intersection. All four plaza tiles lie well within the priest's 25-square (i.e., 26 - 1) limit for walkability, and no roadblocks occur between the temple's walk-start square and any of these plaza tiles, so they become targets for grounded walks. Each of these grounded walks begins with a destination walk to one of the plaza tiles, and then continues for an additional 26 squares further out along the same road. The full quadramble of the priest, therefore, consists of a sequence of walks taking that roamer to the (8 + 26 =) 34th square along the SE road, then the 34th square along the SW road, then the 34th square along the NW road, and then the 34th square along the NE road. The sequence need not begin with the walk to the SE, but the order of the walks will be clockwise as described.

Fig. 2 shows two cases in which both connected and disconnected roads occur within routing zones. The two routing centers of the water supply lying to the SE and SW lie outside the figure and cannot be discussed. The road loop towards the north in the figure lacks a connection to the walk-start square of the water supply, so it is theoretically capable of causing the algorithm to generate default walks for the NW and NE legs of the water carrier's quadramble. The innermost search ring around the NW routing center (Fig. 2) with any roads in it at all contains both disconnected and connected road squares, so the connected road squares will be found before the algorithm's search for road squares halts. The connected road square in the NW zone with the highest search priority is marked by a plaza tile, and it becomes the target for a grounded walk. The destination-mode portion of that NW walk takes the water carrier unerringly through the four-way intersection to the plaza tile, where conversion to random mode for 26 more squares of travel occurs. The algorithm would rather have the walker turn left

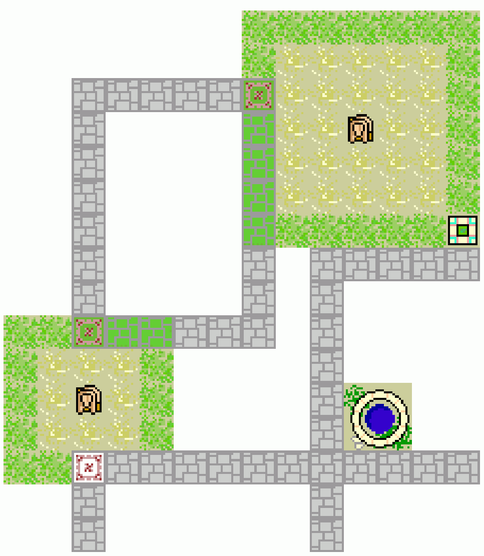

Figure 2. The regions around the northwest (left) and northeast (top) routing centers (small statues) of a water supply examined for roads by the algorithm. Three disconnected road tiles found above the NW routing center fail to halt the search in time to prevent discovery of the walkable plaza tile, which does end the search. Five disconnected road squares near the NE center cause that search to end with the examination of the garden square, so the road just below the NE search area is never found. North lies toward upper left.

than reverse course, so the water carrier continues down the vertical road and off the bottom of the figure.

In the NE routing zone of the water supply (Fig. 2), disconnected road squares appear in the third search ring around the routing center, and connected road squares do not appear until the fourth. The search for roads in the NE routing zone, therefore, halts (since road squares were found in that ring) after examination of the square marked by the garden (the last square of the third search ring to be tested), and the connected road just below the garden is not found. Consequently, the NE leg of the water supply's quadramble defaults out due to disconnection, and the water carrier travels 26 squares from his walk-start square heading east, which takes him off the right side of Fig. 2.

The algorithm's search around the SW routing center of the senet house in Fig. 3 identifies a walk target (green plaza) for a grounded walk just one square below the walk-start square (ordinary plaza) of that building. Another walk target is found around the NE routing center and is marked by a green plaza tile at the top of the figure. This NE target lies 23 squares of travel away from the walk-start square of the senet house, which is within walkable distance for both the labor seeker (a short walker) and the senet player (a medium walker) from that building. On the NE leg of his quadramble, the senet player will make a grounded walk beginning with a destination walk to that green plaza tile. On top of this walk target, conversion to random mode occurs for another 35 squares (***d*** for a medium walker) of travel, which carries the roamer off the top of the figure.

The northwest leg of the quadramble of the senet house in Fig. 3 offers an example of shorting due to distance. In its search around the NW routing center, the algorithm finds a connected road square that could serve as a walk target marked by a green plaza tile at the left edge of the glyphy. This walk target lies 27 squares of travel from the walk-start square of the senet house, which lies beyond the 25-square limit of walkability for the labor seeker (a short walker) from that building. The yellow plaza tile in Fig. 3 shows the furthest location on which a walkable target for the labor seeker could appear. If the senet house labor seeker executes the NW leg of the building's quadramble, he makes a distance-shorted walk. The roaming half of his walk will carry him (via a destination walk) to a go-home transition point on the ordinary road square between the yellow and green plaza tiles. The senet player is a medium walker, and any road square within 34 squares of his walk-start square is walkable. Consequently, on the NW leg

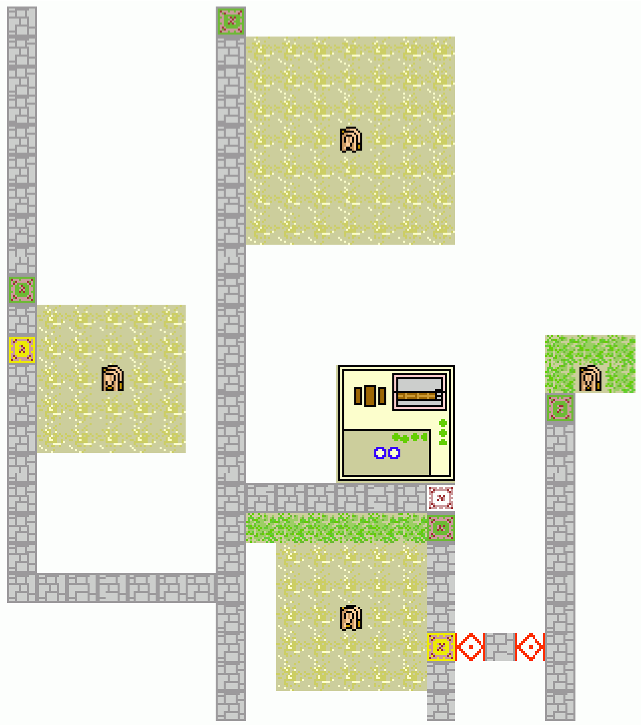

Figure 3. Illustration of grounded walks and walks shorted due to distance and blockage. The areas searched by the algorithm around the four routing centers (small statues) of a senet house are shown by desert squares (searched first) and grassy squares (searched next). Green plaza tiles mark the walk targets.

of his quadramble the senet player makes a grounded walk with the green plaza tile as his walk target. Once he arrives (via destination walk) on that walk target, the senet player travels in random mode for an additional 35 squares (***d***, for a medium walker) up the road and off the figure.

The southeast leg of the senet player's quadramble is shorted due to blockage. In its search around the SE routing center, the algorithm finds the green plaza tile at the top of the road on the right side of Fig. 3. This road square is unwalkable because the shortest circuit connecting it to the walk-start square of the senet house passes through two roadblocks. The blockage-shorted SE leg of the senet player's quadramble begins with a destination walk along the shortest circuit (as if the roamer were trying to reach the walk target) but continues only until the nearest roadblock bars further progress towards the target. On the yellow plaza tile next to the road block, conversion to random mode occurs granting the senet player an additional 30 squares of travel (to keep the total length of the shorted walk equal to ***d*** for this medium walker).

Addenda

This section contains some details and fine-print about roaming-walk generation, which would have distracted from the flow of the development in earlier sections. Some of the information is important enough to have been well researched, but preliminary results are later introduced for a few topics that are hoped to be of interest to readers.

Default details. At present, no prediction can confidently be made about the initial direction of a default walk from a walk-start square located in a three-way intersection, but the series described earlier for corners and straight roads frequently does not apply. From a walk-start square in a three-way intersection, a default walk often makes a “false start”, in which the roamer reverses directions and returns to the intersection after a single square of travel. Neither the direction of the first square of travel, nor the eventual directional choice made after return to the intersection appears to be governed by the default direction-preference sequence for corners and straight roads. The author can, as yet, offer no explanation for how false starts for default walks either from three-way intersections or from dead ends (at which they also sometimes occur) might be generated by the algorithm. Fortunately, default walks of any kind occur infrequently in actual game cities, and default walks originating from three-way intersections are so rare that the author has seen them only in specially constructed walker labs.

Shortest-circuit tie breaking. When two road-connected paths of equal length (and shorter length than all others) exist that connect a building’s walk-start square to a walk target, the algorithm needs a rule to unambiguously identify one of these paths as the shortest circuit to be examined for length and roadblock presence. In addition, the same rule used for choosing between two available shortest circuits (connecting walk start to target) also appears to be used when choosing between two paths of equal length by which a roaming walker could return to his walk-finish square when he re-enters destination mode for the homeward-bound half of his walk. To allow the rule to be described in sufficiently general terms so that it can apply to either situation, I will refer to the square on which the roamer enters destination mode as the walk origin and the square that he is trying to reach as the walk terminus.

Tie-breaking rule. The algorithm selects a destination-mode path for a roaming walker by starting at the walk terminus and looking backwards along the path towards the walk origin. When two alternative paths of equal length (shorter than all others) can be traced from the walk terminus to the walk origin, the algorithm chooses between the two paths based on the directions they take from the point at which they diverge, according to the following order of preference: NE > SE > SW > NW.

The order of precendence described in the rule, above, agrees exactly with the order described in the first edition of Ambulomancy. The only difference between the original and amended versions of the rule is that the new version states that the paths must be explored backwards from terminus to origin, whereas the original version said that the algorithm looked for the divergence point beginning at the walk-start square and heading in the outward-bound direction.

Fig. 4 shows a situation that requires the algorithm to apply its tie-breaking rule to a road geometry that reveals its use of the backward-looking version of the rule. When the dentist is dispatched on a walk of any of the four formal directions, his target (one of the ordinary plaza tiles in the figure) will be “unwalkable”. When the roadblock is present, it lies on the dentist's path between his walk-start square and all of his four targets, so all four of his walks short out due to blockage. If the roadblock is removed, then all of the dentist's walks short out due to distance, because the closest of his targets can only be reached after 39 squares of travel - well beyond the limit of 25 for a short walkers, like a dentist. The dentist's choice between the yellow and blue paths shown in the Fig. 4 is the same whether the roadblock is present or not, so the following discussion will explore the situation more commonly encountered in actual cities - shorting due to blockage (with the roadblock installed).

When the roadblock is present, all four of the dentist's walks in Fig. 4 pass through three phases: outward-bound destination mode from the walk-start square to the square just SW of (or below, in the figure) the roadblock; then in random mode to the green plaza square; and finally in destination mode to return from the green plaza square to the walk-finish square of the dentist's office. The tie-breaking rule applies to both of the destination-mode phases. In the outward-bound destination mode walk, the walk terminus is the road square touching the roadblock from its southwest, because that is the last road square the dentist can legally (without violating the roadblock) reach in trying to get from his start to his target. The amended tie-breaking rule is applied by starting at this terminus and working backward toward the dentist's walk-start square looking for the point of divergence, which in this case is the red road square. From this divergence point, the first yellow road square lies SW, and the first blue square lies NW. The algorithm seems to look for roads around a divergence point in the same order that it looks for roads around a 1x1 building when it is trying to identify a walk-start square, so the SW branch is chosen. Once that selection has been made, the road connection to the walk-start square of the dentist's office becomes unambiguous, and the path from terminus to origin (including the yellow squares) has been chosen. The dentist is then sent along that path in the reverse order, from origin (walk start) to terminus.

At the terminus (next to the roadblock), the dentist switches to random mode to use up the remaining squares of his 26-square outward-bound walk. This takes him to the green plaza. On the green plaza (his homeward origin) he reverts to destination mode for the return trip to his walk-finish square (his homeward terminus). Once again, the dentist has two alternative routes of equal length leading to his terminus, so the algorithm begins tracing the route from the walk-finish square (terminus) to the green plaza (origin) looking for the divergence point, which is found to be the green road square in Fig. 4. From this divergence point, the yellow road square lies SE and the blue square lies NE. NE trumps SE, and the blue path is selected. The dentist, thus, returns from the green plaza square to his walk finish via the blue road rather than the yellow road.

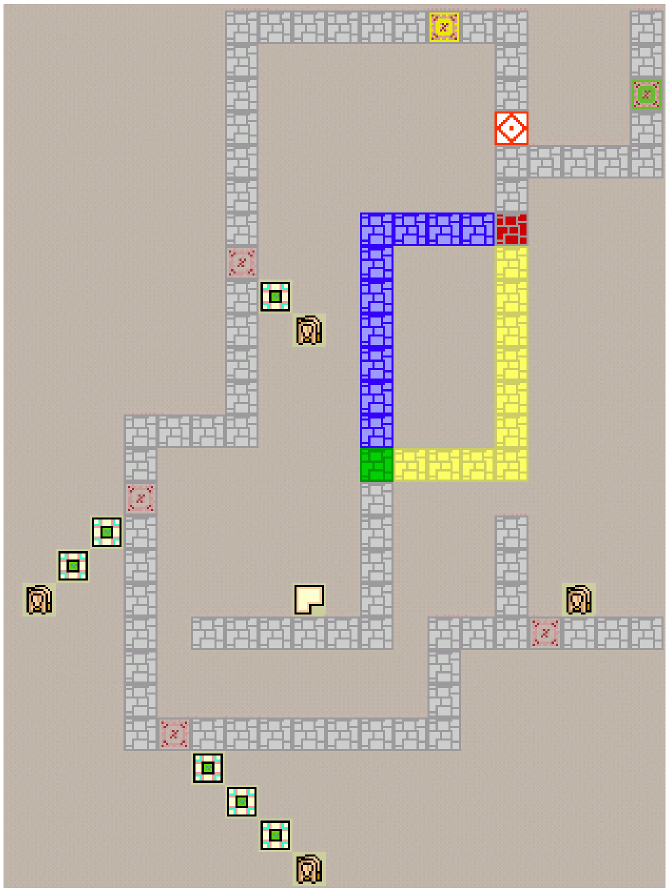

Figure 4. The tie-breaking rule in action.

When the roadblock is removed, all the dentist's walks become distance shorted and lack a random-mode phase separating the outward- and homeward-bound phases. The yellow plaza serves both as his outward-bound walk terminus and as his homeward-bound walk origin. His path choices remain the same as they were when his walks were blockage-shorted: yellow path for the outward-bound walk, and blue path for the homeward-bound walk.

Off-map squares. Roamer-dispatching buildings located near map edges often extend portions of their routing zones or even routing centers beyond the edges of the map. The algorithm analyzes road systems for roaming-walk generation as if the map were surrounded by enough empty squares to contain these zones and centers. The algorithm appears to add no surprises to these off-map squares, such as invisible disconnected roads. Instead, its search of off-map squares simply finds no roads until the usual search order leads it back onto the map to explore visible squares on which the player may have placed roads. Readers are cautioned, however, that the author has not been clever enough to determine whether the kingdom road at a map corner extends indefinitely directly away from the map center.

Water. Streams and the Nile offer no barrier to the roamer algorithm's search for road squares. It finds road squares on the other side of water. Bridges are treated like any other road squares and walk targets can be found along them.

The effect of ferries on roaming walk generation have not yet been fully explored. However, some preliminary results are available. A Ferry route appears to provide a blocked connection between roads on opposite banks, but the algorithm does not treat ferry routes exactly as if they were roads with roadblocks on the river banks at either end. When the algorithm's search around a routing center finds roads on the opposite bank of a river from the building being analyzed, a ferry route causes those roads to be treated as connected, although they are certainly unwalkable, since roaming walkers may not use ferries.

Fig. 5 illustrates the effect of a ferry on the southwest leg of a physician's quadramble. The SW routing center (small statue) falls on the opposite bank of the river from the physician's office. If the road on the opposite bank, the ferry, the bridge (symbolized by plaza tiles), and the roadblocks at either end of the bridge are all removed, the algorithm finds the green plaza tile on the road leading to the left as a target for a grounded SW walk by the physician. If only the road on the opposite bank is added (but not the ferry or bridge), that disconnected road is found by the algorithm and the SW leg of the physician's office defaults out. In this case, the physician makes a 26-square random default walk up the road toward the top of the figure (towards the NE). If the bridge and its roadblocks (but not the ferry) are added to the picture, the algorithm identifies the yellow plaza tile on the opposite bank as a target rendered unwalkable due to blockage. In this case, the physician makes a blockage-shorted walk that takes him to the road square next to the nearest road block (in destination

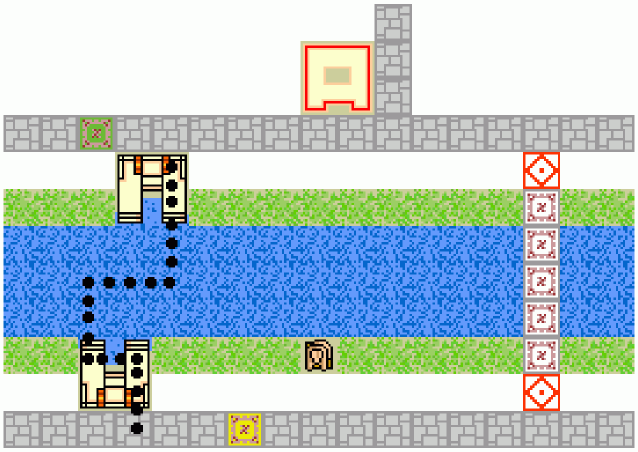

Figure 5. A physician's office with its SW routing center across a river. Colored plaza tiles show SW walk targets of the physician's office found in the presence (yellow) or absence (green) of the road on the far bank. The five plaza tiles represent a bridge. North lies toward upper left.

mode) and then further along the road to the right in random mode until he reaches his go-home point 26 squares from his walk start. If only the ferry (but not the bridge or its roadblocks) is added to connect the road on the opposite bank to the roads traveled by the physician, the same yellow plaza tile is identified as a target which is again deemed blocked and unwalkable because travel by ferry is prohibited for roamers. The physician's SW quadramble leg is affected by the ferry's presence, however, since that walk does not take the physician NE (as when the leg was defaulted out) or SE (as when the leg was shorted out by the bridge plus roadblocks). Instead, the physician makes a 26-square walk to the left, past the ferry landing. This is clearly a case of shorting due to blockage even though no physical roadblocks are present.

Until recently, the paths followed by ferries and immigrants between ferry landings were regarded as subjects lying outside the province of roaming walker research. However, it was possible in a lab map to get some idea of path length calculated by the algorithm for the ferry route by watching evicted homeless travel between ferry landings. They followed the twisted path shown by the dotted line in Fig. 5.

A short circuit traced from the physician's walk-start square through the ferry route (along the dotted path in Fig. 5) to the yellow plaza-marked walk target has a length of 22 squares. If all the elements shown in Fig. 5 are present on the map, then an alternative short circuit with length of 22 squares can also be traced along the route provided by the bridge. The glyphy, thus, shows a situation requiring the shortest-circuit tie-breaking rule. The physician reaches the point of divergence between the two paths at the three-way intersection. As the rule promises, his blockage-shorted SW walk takes him towards the bridge instead of the ferry, since the initial direction from the intersection along that path takes him SE which is preferred (under the rule) over the NW direction that he would need to take to follow the short circuit to the ferry. If the entire bridge (and its roadblocks) are moved one square to the right from the position shown in Fig. 5, then the length of the short circuit through the bridge increases to 24 squares. The circuit through the ferry (with length 22 squares) now unambiguously becomes the shortest circuit. In this case, the physician turns NW at the intersection and passes the ferry on his SW walk.

A few additional results about ferries are available. The blocked connection between roads provided by a ferry is lost if either landing loses its staff. In the author's tests, the algorithm has consistently refused to "find" ferry path squares in a routing zone as though they were road squares for use as walk targets. Thus, ferry paths act sufficiently like roads to permit shorting due to blockage, but not enough like roads to contain walk targets.

Initial seeker invariance. This rule is useful when you are trying to put two or more fireman "out of phase" in a multiple block structure like a figure-8 block. When a building is first placed on a particular map square, we have (to date) no way of predicting in advance which walk of the four in its quadramble will be the first to be executed by the labor seeker from that building. However, if you delete that building and install any other building on top of the original structure's position so that their northernmost squares fell/fall on the same map square, then the initial labor seeker dispatched by the second building will execute a first walk with the same formal direction (e.g., the northeast walk) that the previous building's seeker followed on his first walk. This rule applies even if radical changes are made to the roads between the original building's destruction and the second building's installation. The first walks may be markedly different, but their formal semicardinal directions will be the same.

Seeker suppression. Most service-providing roamers recruit labor, so their home buildings do not need to dispatch any labor seekers after the first one, provided that the service roamer passes "ample" housing on every one of his walks. In contrast, the magistrate and policeman do not recruit labor, and their buildings will continue to emit labor-seeking "citizens" even in housing blocks. However, one trip past abundant housing by the labor seeker from a courthouse or police station appears to collect enough "job applications" so that the service roamer can make six to eight walks before the labor seeker must appear again.

Quadramble sharing. A service-providing building's labor seeker and its service-providing roamer share the same quadramble. If the seeker that appears following building construction executes a quadramble leg with the formal direction northeast, the service-provider that appears next will follow the southeast walk in the same quadramble, i.e., the next walk in clockwise order. If a building becomes dissatisfied with the amount of housing its service roamer encountered during his last walk (or a city-wide labor shortage develops), then it will emit a labor seeker (often just a few frames) before it dispatches the service roamer again. Once again, the labor seeker uses up the next walk in the building's quadramble, which causes the service roamer to skip that walk.

In a rectangular housing block with a small (e.g., 48-square or less) loop, skipping a single walk usually poses no threat to block stability. But, if the roamer provides coverage to an exotic multi-block structure (like a windmill) that provides some houses with only one service walk per quadramble for some vital roamer, skipping that one walk can have dramatic consequences.

Synchronized firemen. This property of firemen probably has parallels with most other roamers, but I have only found a use for it when trying to get two fireman to always circle a block going in the same direction and to stay on opposite sides of the block. Very useful for fire suppression at very hard difficulty.

Firemen can only emerge from their firehouses at certain, regularly repeating start times. If time in Pharaoh were measured in quarter notes, firemen could be said to appear on their start squares only on the downbeats of a four-four measure. This means that if you place two firehouses simultaneously on the map, not only will the first firemen generated by those structures appear simultaneously, but the second and all subsequent firemen generated by the same firehouses may also appear simultaneously, if certain conditions are satisfied. It is not necessary for the firemen to always walk exactly the same paths or for the paths for the two firehouses to be of exactly equal length. All that is necessary, is that all the walks followed by all the firemen from both firehouses must have lengths belonging to the same synchronization group.

Here are the synchronization groups for the path lengths (measured from walk start to walk finish and counting the walk start as square 0) of firemen:

Group 43:	43, 44, 43, 46
Group 47:	47, 48, 49
Group 50:	50, 51,52, 53
Group 54:	54, 55, 56
Group 57:	57, 58, 59
Group 60:	60, 61, 62, 63
Group 64:	64, 65, 66
Group 67:	67, 68, 69, 70

I am sure Groups 71 and higher exist, but the would not be terribly useful since housing blocks are unlikely to have main loops that long.

Teleportation from pavilions

This section originally appeared as a separate thread in the Pharaoh Game Help Forum titled "Walkers and teleporters from pavilions". I repeat the information here to provide the reader with as complete a set of tools (in a single package) as I can to support the walk-engineering of housing blocks. This section focuses on teleportation: the sudden appearance of a walker on a square not adjacent to the building that dispatched him (or her). Teleportation has to qualify as a pretty specialized subject within the general area of walker studies, so readers should expect much of the material below to sound like complete gibberish unless they have studied the earlier sections on roaming walker behavior in this document.

Sometimes, the entertainers generated by pavilions really wind up in the weeds - completely off the road. Although the discussion below describes in general terms the conditions that produce such behavior, a full treatment of off-roading by entertainers and architects (and possibly others?) will have to wait for a separate post devoted to that subject.

Critical squares. All three kinds of entertainers treat a pavilion as if it consisted of nothing but the 2x2 dance stage. The locations of the music and juggle stages are ignored. The walk-start and -finish squares of the dance stage (and therefore the entire pavilion) can be determined as Brugle describes for common walkers from 2x2 buildings in his thread on the Pharaoh Game Help Topics forum entitled "Random walker start and finish points". Entertainers arriving from their schools as destination walkers are a bit trickier. If the stage for his/her type of arriving walker is unoccupied, then the walker disappears on the walk-start square of the dance stage. If a show is in progress on the stage for entertainers of the new arrival's type, then the walker disappears on the walk-finish square of the dance stage. The north square of the dance stage also serves as the "governing square" from which the locations of the four routing centers can be determined by the usual procedures outlined in Ambulomancy: eight squares away in all four semicardinal directions.

Teleporting entertainers. Of the three kinds of entertainers dispatched by pavilions, musicians and dancers teleport, but jugglers do not. However, jugglers, as we will see, do play a pivotal role in facilitating teleportation. Like the other entertainers in Pharaoh, pavilion-dispatched entertainers are medium walkers with 35-square default walks. Like a bandstand (See the "Bandstand walkers" thread.), a pavilion has only a single quadramble which is shared by all of its walkers (entertainers and the labor recruiter). Each roaming walker dispatched by the building is assigned the next formal direction in a single, clockwise cycle of walks.

Initiation of teleportation. Starting with the first game of Pharaoh most builders ever played, we all noticed that a small amount of time passes between the instant a roaming walker who has just returned to his building disappears on his walk-finish square and when he or she reappears on his walk-start square to begin his next walk. I call that interval the walker's coffee break. Teleportation from a pavilion will not happen unless two entertainers are simultaneously taking a coffee break. Teleportation is not guaranteed to happen if a second entertainer begins his coffee break only an animation frame or two before the first entertainer was scheduled to end his break; a certain amount of time (not yet precisely measured) together in the break room seems to be required to trigger teleportation, although as best I can tell it is not even as long as it takes a walker to advance a single square of travel. In the discussions below, when time becomes important, I quantify time in terms of "squares of travel" (SoT), because I measured time using a modification of Jimhotep's clock described in his "Pharaoh Calender" thread.

Jugglers as teleportation catalysts. The game has a trick both to increase and to decrease the likelihood of simultaneous coffee breaks (and, therefore, teleportation): changing coffee-break duration. Long breaks increase the probability that two walkers will have overlapping breaks. The principal way that the presence of jugglers stimulate teleportation is by increasing break length. When the juggle stage is unoccupied, musicians and dancers take short breaks, usually lasting just two to four SoT. While a show is in progress on the juggle stage, all the pavilion's walkers take long, selfindulgent coffee breaks usually lasting 22 to 24 SoT.

Keeping breaks short is not the game's only method for inhibiting teleportation in the absence of jugglers. If a musician and dancer arrive in fairly close succession (but not simultaneously) at a juggle-free pavilion, the first arriving walker gets yanked out of the break room and booted out onto the street as fast as possible. The second walker is then detained for a fairly long (for a juggle-free pavilion) break of 10 to 12 SoT. This difference in break time substantially increases the separation between the arrival times of these two entertainers when they return to the pavilion.

Despite the algorithm's efforts to decrease the likelihood of teleportation in the absence of jugglers, teleportation from pavilions does occur without jugglers. For this to happen, however, a dancer and a musician must disappear on the walk-finish square of their pavilion at almost the same time. A two to three animation frame difference in arrival time is close enough to simultaneity to permit teleportation, but a two to three SoT difference in arrival times is too great to allow teleportation. Like a chemical reaction with a high activation energy, teleportation is accelerated by the presence of an appropriate catalyst (jugglers) but will proceed slowly without the catalyst.

Jump teams. When two or three walkers have been on break together long enough for teleportation to happen they form a "jump team". One walker in each jump team (hereafter called the "spotter" or non-jumper) does not teleport and all the other team members do. If the jump team includes a juggler, he will be the spotter. Otherwise, the musician spots for the dancer.

All the members of a jump team seem to the player to appear simultaneously on their walk-start squares. I have been unable to detect so much as a single animation frame of difference in their times of appearance. Nevertheless, the algorithm clearly assigns them a formal order of appearance which I have never observed to vary: Jugglers appear first, then musicians, and finally dancers. The order of appearance does *not* depend upon the order of arrival of the team members at the pavilion (while a team is assembling in the break room); the order depends only on occupations.

Remote walk starts. When the jump team is dispatched by the pavilion, the first team member (usually the juggler) appears on the pavilion's usual walk-start square and is dispatched on a walk with whatever formal direction (e.g., NE) is the next one due for execution in the pavilion's clockwise cycle of walks (quadramble). Other members of the jump team are assigned later walks in the pavilion's quadramble in the order of their (closely timed) appearances. Thus, for a three-man jump team, if the juggler is assigned a NE walk, the musician will be given the SE walk, and the dancer will get the SW walk. Unlike the juggler, who uses the pavilion's usual walk-start square, the musician and dancer in a three-man team use non-standard, teleporting walk-start squares. In this example, the musician will use the walk-target of the juggler as her walk-start square, and the dancer will use the walk-target of the musician as a walk-start square. The use of prior walk-targets as current walk-start squares certainly gives the appearance of teleportation, since the musician and dancer suddenly wink into existence up to 14 squares away from the north square of the dance stage.

Although they begin on unusual squares, the walks of musicians and dancers following teleportation behave in many ways like the walks of other roamers in Pharaoh, allowing for differences in walk length (since entertainers, including senet players and zoo keepers, are medium walkers and nobody else is, except tax collectors). Unfortunately, the other walkers in Pharaoh whom teleporters from pavilions most closely resemble are walkers with “bad habits” like tax collectors (who vanish at the end of some of their walks rather than returning, visibly to their walk-finish squares) and architects (who not only sometimes vanish, but on other occasions wander right off the roads). Senet players have also been reported to teleport. Fortunately, the analytical tools needed to understand and predict the behavior of pavilion roamers after teleportation are ones that apply to all roaming walkers (routing centers, road-search order, walk targets, destination- and random-mode portions of walks, etc.) and that have already been described above. The remainder of this section explores the various kinds of roaming walks that entertainers from pavilions execute.

Grounded walks. Grounded walks for spotters and teleporters follow all the usual rules. If the shortest-road connected path between walk-start square (traditional or teleport-style) and walk target has length of 34 (i.e., *d* - 1) squares or less and contains no roadblock, the entertainer’s walk follows that path (in destination mode) and converts to random mode (for 35 more squares of travel) without reversing course on top of the walk target.

Illusory counter-clockwise rotation. In the first (March 2005) edition of Ambulomancy (“Predicting Roaming Walks”), I stated that teleporting entertainers appeared on “landing” squares in a counterclockwise rotation. This observation appeared to violate the rule applying to all other walkers that says their walks must progress in a clockwise rotation of formal directions. The counterclockwise rotation is now revealed as an artifact.

Observations of individual entertainers leaving pavilions from the usual walk-start point and two- and three-man jump teams have clearly revealed that all the roamers from a pavilion draw their walks from a single queue and the walks within that queue are arranged in a clockwise progression of formal directions. Fig. 6 provides the starting point for an example. Even though the order is imperceptible to the player, the juggler is assigned his northeast walk first (targeting the yellow plaza), then the musician gets the next (SE) walk (starting on yellow plaza, targeting the pink one), and finally the dancer begins a SW walk (to the green plaza from the pink). All three entertainers will return to the walk-finish square of the pavilion at more or less the same time - close enough to permit teleportation. Therefore, they will all leave the pavilion on their next walks as a three-man jump group.

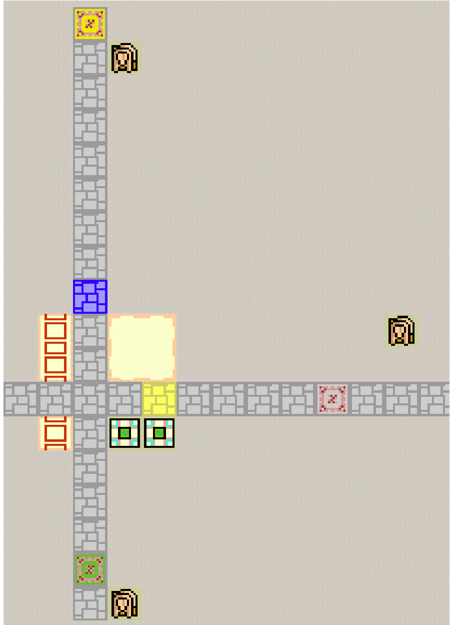

Fig. 6. Critical squares of a pavilion, including blue walk finish, yellow non-teleporting walk start and plaza-marked teleporting walk starts.

The next walk available, after the targets shown in Fig.1 have been used, has a formal direction of NW, so the juggler (in the second round of teleportation) will appear first as the spotter for the group on the pavilion’s usual walk-start square, and begin a grounded walk with a NW target. The musician will teleport to a remote walk-start square (the juggler’s target) for a grounded walk to a NE target (the same one shown in Fig. 1 as a yellow plaza). At the same apparent time (but actually last of the three), the dancer will teleport to the musician's walk target (now the yellow plaza square) and begin a grounded walk to a SE target (the ordinary plaza square in Fig. 1). To the player following only the dancer, the teleportation-landing (or remote walk-start) squares appear to be used by the dancer in counterclockwise order (e.g., a SE landing in Fig. 7, then a NE landing for the second round of walks). The same thing happens if the player watches only the musician. When all four of the walks in the pavilion's quadramble are grounded walks of similar length, this apparent rotation can continue indefinitely, as long as all the entertainers leave the pavilion as part of three-man jump groups. Thus, I perceived a sequence of 270-degree clockwise rotations (for individual performers) as a series of 90-degree counterclockwise rotations. Now, we know better.

Shorted walks. The designers evidently changed the way the walker algorithm handles shorted walks for pavilion entertainers (and for architects) from the way those walks are described in Ambulomancy for the majority of walkers. Instead of just two kinds of walks that can be produced either by roadblocks on, or by excessive lengths of, the shortest circuit, these walkers have at least three. One of the three kinds of walks can send the roamers off road, and (as mentioned earlier), is still the subject of on-going research. In this work, I content myself with approximately describing the conditions that can trigger off-roading.

In the following discussion, I once again use the letter *d* for the length of a default walk: 35 squares for entertainers; 43 squares for architects.

Walks shorted only by blockage. If the distance of the shortest circuit from a roamers's walk-start square (next to the building of origin or teleporting) to the target of a walk is *d* - 1 squares or less and a roadblock lies on that path, walks of ordinary roamers short out for reasons that can only be attributed to blockage; without the roadblock, no shorting would have occurred. The outward-bound halves of the walks followed by entertainers and architects under these circumstances look exactly like blockage-shorted walks for ordinary roamers. Unlike most walkers, pavilion entertainers and architects vanish at the end of the outward-bound half of their only-blockage-shorted walks, i.e., after *d* squares of travel from the walk start.

When architects vanish at the end of a walk shorted only by blockage, they ghost home invisibly by the shortest available route even if that route takes them off road. They do not show up on the usual game display, but their presence and movement can be detected by passing the footprint of a large building (as if the player was about to position that building) above an off-road portion of the shortest path between the architect's vanishing point and his office. Ghosting architects travel off road and (probably) diagonally. Their ghosts appear to travel in a series of steps rather than in a smooth diagonal, but they arrive at their destinations much faster than they could if they were traveling only in NE-SW and NW-SE directions.

In tests of pavilions with just one kind of entertainer, the roamers usually reappear on their next walks just two to four SoT after vanishing at the end of walks shorted only by blockage. Thus, sufficient time was not available for the entertainer to have walked invisibly back to the pavilion. When an entertainer, who is part of a jump team, disappears after a blocked-only short, too little time also passes in many cases for the walker to ghost back to the pavilion. Also, no ghosts of pavilion entertainers have been detected between their vanishing points and their walk-finish points. Thus, entertainers from pavilions do not appear to ghost, unlike architects.

Walks aborted by distance. If a walk in a pavilion's quadramble assigns an entertainer an unobstructed target lying 35 (*d*) squares or more from the roamers walk-start (traditional or teleported), the entertainer's walk does not short out by having its length reduced to *d* squares (like those of ordinary walkers). Instead, the entertainer simply walks all the way to the target and vanishes, thus appearing to abort the walk at its midpoint without returning to his walk finish. I do not know if there is a absolute upper limit to the length of this walk, but I do know the entertainer will travel a lot further than 35 squares. If a roadblock lies further than *d* squares from the walk-start point along the shortest road-connected path to the target, the entertainer flouts the roadblock by passing through it. If a road-block appears earlier, a confused walk is likely to be generated (See below.).

Walks confused by blockage and distance. If the shortest road-connected path to the entertainer's target is further than *d* squares from the walk start (so aborting due to distance would be possible without a roadblock) and the nearest roadblock on that path is closer than *d* squares (so shorting due to blockage would have been possible for other walkers), a unusual type of walk often results that appears to be possible for both pavilion entertainers and architects (and probably other roamers, like tax collectors, as Brugle notes in reply 6 to "Walkers and teleporters from pavilions", and I have subsequently confirmed independently): the confused walk.

Several beautiful examples of confused walks by an architect were posted by Max as reply 4 to "Predicting Roaming Walks". The column labeled "walk-4" contains a mixture of distance aborted walks (when rbc at 44 or 60), only-blockage-shorted walks (when rbc is at 1 or 15), and confused walks (rbc's from 16 to 43). The confused walks in Max's table begin like blockage-shorted walks, but on their *d*-th square, they seem to switch into a new walk mode (for 2 to 46 squares of travel) in a strange middle-phase of their walk: neither outward- nor homeward-bound. In many of Max's examples, the middle phase of the architect's walk carries him off the roads. In others, the architect appears to meekly return part of the way (by road) toward his office before vanishing. I have been able to reproduce the confused walks that Max reported using pavilion entertainers instead of architects (allowing for minor adjustments need to accommodate the difference between their endurance classes). In addition, I have been able to create off-road confused walks by teleporters from pavilions provided that their shortest circuits meet the same requirements as for traditional start-square users: distance to target > *d*; distance to closest intervening roadblock < *d*. To date, I have failed to figure out how to predict confused walks. Therefore, in housing block design, I only rely on the first *d* squares of travel by a confusable walker on a quadramble leg satisfying the conditions necessary to generate confusion. If the confused portion of a walk has been seen (during early block development when not all buildings are in place) to carry the walker off road, placing a building in his/her way to obstruct departure from the road will cause the walker to vanish and quickly be available for reappearance on the walk start square. This recycles the walker much more quickly than if it was allowed to good wandering about in the weeds off road.

Disconnected walk targets. The default walks of ordinary walkers from traditional walk-start squares look the same whether the walk was defaulted by road absence or by disconnection. When entertainers teleport from a pavilion with a defaulted walk in its quadramble, however, their behavior reveals the cause of defaulting unambiguously. The behavior of teleporting entertainers makes it possible to be more precise (than I was above in "Absent and disconnected roads") when describing how disconnected road squares default out a quadramble leg. As the roaming-walker algorithm searches (in the order of priorities shown in Table 1) the region around a shorting center it usually discovers a road square. The first road square found during the search becomes a provisional walk target. The algorithm then checks for a road connection to the affected building's walk-start square. If a connection is found, the walk target's status is changed from provisional to confirmed and the target assignment portion of the algorithm ends. If no road connection is found to the provisional target, the algorithm continues searching the remainder of the search ring (all the squares at the same distance from the routing center as the provisional target) "in hopes" of finding a connected road. As additional road squares are found, they are each checked for a road connection to the walk-start square. As soon as a connected road square (in the same search ring) is found, it becomes the confirmed walk target and target assignment ends. If the search of all the squares in the same ring as the provisional (disconnected) target finishes without discovering any connected road squares, then the status of the provisional walk target is changed to confirmed and targeting ends.

The greater detail offered by this new description of the targeting process specifies the location of a disconnected road square as the target for a defaulted walk. Although any walker that the algorithm tries to aim at that disconnected target from the usual walk start square cannot get there, a teleporting entertainer can use that disconnected target as a starting (or "landing") square.

Walks defaulted by disconnection. When the roaming-walker algorithm cannot trace a road connection from the walk-start square to the target, the corresponding walk in a building's quadramble defaults out. After appearing on the walk-start square (which may be reached by teleportation) the walker executes a random-mode walk of default length (35 squares for medium-walking entertainers) with the usual default initial directions that apply to default walks beginning on straight roads or corners.

At the end of a default walk, the algorithm sends the entertainer on a destination walk back to the walk-finish square of the Pavilion by the shortest road-connected route. If no road connected path exists, then the entertainer vanishes and immediately is treated as being on a coffee break. Naturally, entertainers who have teleported onto disconnected walk-start squares are going to have a tough time tracing a road connection back to the pavilion.

Walks defaulted by road absence. When the usual procedures of the algorithm find a road-square to use as a walk-start point but cannot find a road in the search zone around the relevant routing center to use as a walk target, the corresponding walk of the pavilion (or any other building) defaults out. However, the algorithm may not be able to find a legitimate walk-start square, i.e., one with a road on it. This situation only occurs for teleporters, since the pavilion must touch a road suitable for use as a traditional walk-start square or the pavilion's labor recruiter could not emerge to find the workers needed to operate the pavilion. Teleporters are different. They use the walk-target of the previous walk in the pavilion's quadramble as their walk-start squares, and a walk target does not have to exist. Clearly, the algorithm needs a way to handle a situation in which an entertainer is ready to teleport from a pavilion, but no walk-start square can be found for him/her because the previous walk was a walk defaulted due to road absence.

When the roaming walker algorithm cannot use the walk-target of the previous walk as the walk-start square for a teleported walk, the algorithm puts the affected entertainer on an "emergency walk-start square": the #1 priority search square of the routing center for the walk *before* the *previous* walk, and tries to use that square as a walk-start point for the entertainer. Thus, a musician who is scheduled for a teleportation-begun walk to the NE from a pavilion with a defaulted NW walk (i.e., with no walk target for its NW walks, and no ordinary teleportation walk-start squares for its NE walks), will appear on the emergency walk-start square one square due north from the SW routing center. If a road exists on the emergency walk-start square, the walker can execute any kind of walk (grounded, defaulted, shorted) from it under the usual rules applicable to the local road geometry. However, no road square is required on the emergency walk-start square for an entertainer to appear there. If an entertainer appears on the emergency walk-start square and that square lacks a road, the entertainer vanishes after a very few animation frames of time have passed.

If the emergency walk-start square of a dancer in a three-man jump team contains a road, that square will be simultaneously serving as the ordinary teleporting walk-start square of the musician. The algorithm starts both entertainers on that same square at the same apparent time for the walks. The two walks are likely to be somewhat different, however. The musician will be on a walk that has been defaulted out due to road absence. She may, however, be able to trace a connection to the walk-finish square of the pavilion, in which case she will visibly walk there (in default mode) at the end of her 35-squares in random mode. The dancer may or may not have a walk target around the pavilion's NE routing center, and that shortest (road-connected) route to that target may or may not be roadblocked. The dancer can, therefore, execute nearly any conceivable walk from the emergency walk-start square. Whether the dancer's emergency walk start contains or lacks a road, and whether she makes a walk or disappears in a animation frame or two, her appearance on the emergency square uses a walk from the pavilion's queue and advances the formal direction for the next walk by 90 degrees clockwise.

Teleportation from a pavilion with a walk defaulted due to road absence can be difficult to visualize from a description, so Fig. 7 shows an example. Fig. 7 illustrates the critical squares that apply to the walks to be executed by the members of a three-man jump team who are about to end their coffee break from the pavilion shown. Acting as the non-jumping spotter, the juggler appears on the usual walk-start square of the pavilion (yellow road in Fig. 7) to begin a walk with a formal direction (in this example) of southwest. The musician begins her NW walk by appearing on her primary teleporting walk-start square (yellow plaza), which in this situation is the same square that the juggler uses as the target of his SW walk. The musician has no target for her NW walk, since no road squares appear in the pavilion's NW routing zone (desert squares), so she will be making a default walk. The dancer would normally use the musician's NW walk target as her primary teleporting walk-start square, but the musician has defaulted due to a lack of a NW target. Consequently, the dancer resorts to the emergency walk-start square for a walk with a formal direction of NE which falls on the #1 search priority square near the SW routing center and is the same yellow

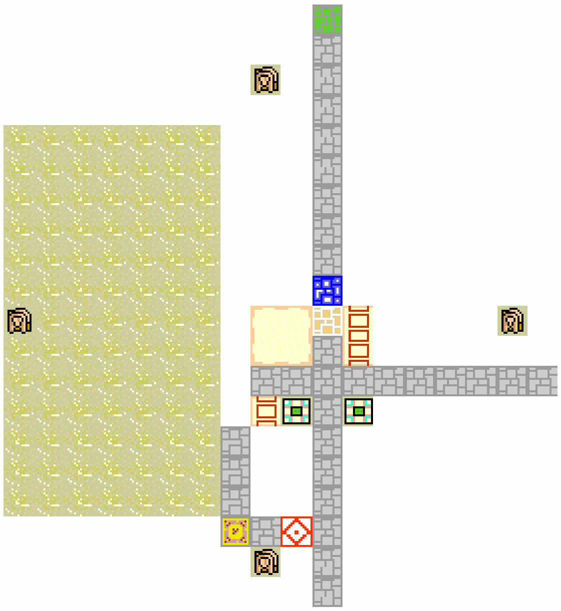

Fig. 7. The absence of a road square in the NW routing zone deprives a musician on a NW walk of a target and a dancer on a NE walk of an ordinary teleporting walk start square. The dancer uses the yellow plaza as her emergence walk start.

plaza tile the musician uses as her primary teleporting walk start. The musician and dancer appear on the yellow plaza tile simultaneously and the game superimposes their images.

The dancer and the musician do not execute the same walks. The musician (on a default walk) leaves the yellow plaza tile heading towards the NE (the default direction for a west corner). Meanwhile, the dancer, who uses the green plaza at the top of Fig. 7 as a walk target, shorts out due to the presence of the roadblock on the shortest route to that target. On this blockage-shorted walk, the dancer traces as much of the route to her target (one square of travel) as possible before the roadblock prevents further progress, and then completes the rest of her (total length = 35 squares) shorted walk trapped in the little piece of road to the left of the roadblock wandering aimlessly in random mode. Thus, the dancer moves SE on the first square of travel, so a difference between the walks of the musician and dancer reveals itself immediately. The juggler's SW walk also shorts out due to blockage because the same roadblock that stopped the dancer also gets between his walk-start (blue road) and his target (yellow plaza). The juggler, therefore, proceeds in destination mode straight along the route to the yellow plaza square as far as he can without hitting the roadblock, and then converts to random mode (without reversing course) to use up the remainder of his 35-square shorted walk.

At the end of their blockage-shorted walks, the juggler and dancer both vanish and return to the pavilion immediately to start a coffee break. The musician does not vanish, because she is on a default walk and has a road-connected path to the pavilion's walk-finish square. At the end of random mode, she converts to destination mode to get to her walk finish, which lets her pass through the roadblock.

At the end of the Applied Ambulomancy section below, several roaming walks made by entertainers from a pavilion are analyzed including both teleporting and non-teleporting confused walks.

Discussion

An understanding of the algorithm used by Pharaoh to generate roaming walks can affect city design in ways that vary from modest to profound. At the very least, we now know enough not to make connections to external roads through a roadblock from a stub road included within a housing block to hold a venue. Any service-providing buildings in the block that found walk targets on external roads connected to that roadblock would shoot their roamers (on the affected blockage-shorted walks) straight into the stub road, which could cause the usual failures to loop that we have grown to hate. Players who can stomach a bit of square counting may find themselves celebrating (rather than grudgingly tolerating) three-way and four way intersections by incorporating them into compound housing block designs (like two blocks fused through a four-way intersection to make a figure-eight) that permit one block's worth of service buildings to support two or more block's worth of housing. Massive industrial blocks can be kept fully staffed if blocked shorting is used to force the labor seekers to head toward the housing instead of wandering about aimlessly. More applications are out there waiting for us to develop. I hope you have as much fun predicting and controlling roaming walkers as I have.

Researchers suffer from a seemingly irresistible need to coin Latin and Greek jargon for the objects of their study, and I find myself susceptible to this compulsion We now have a body of knowledge that we could call simply (and far too intelligibly to be satisfying) "roaming walker theory" or "roamerology", but where would the fun be in that? Instead, I hope that readers will not begrudge the author's little reward to himself for his time and trouble in developing the information related above: a new word for this field of study. For this purpose, we can reach again for the same Latin word "ambulare" (to walk) that I used years ago in manufacturing the term quadramble and combine it with the“-mancy” that appears in “necromancy” (Middle English, from Old French -mancie, from Late Latin -manta, from Greek manteia, -manteia, from manteuesthai, *to prophesy*, from mantis, *prophet*) to produce a delightful little word for the prediction or “divination of walking” and the same word that began this article: Ambulomancy.

Applied Ambulomancy

In this section, I explore an example of a housing block that was stable in one of my labs at very hard difficulty for several game decades. Analyzing the quadrambles of six of the roaming walker-dispatching buildings that keep this block running smoothly offers, I think, a much more effective way of learning to practice ambulomancy than reading a bunch of sterile rules affords.

The housing block to be analyzed (Fig. 8) exhibits a few desirable properties. To my way of thinking, the most attractive feature of the block is that it is as nearly fireproof as I can devise. (More on this below.) The block also provides intersections to support two pavilions and a band stand. Additional corners can hold three juggle platforms, although the one that should have been under the physician is missing from Fig. 8. Despite the presence of two stub roads and a little branched alley extending from the main 52- square loop, there are at least six positions for 2x2 buildings whose roaming walkers would be certain to loop the block on all four legs of their quadrambles. In Fig. 8, the positions are shown occupied by five water supplies and a physician’s office. At least five other positions for 2x2 buildings are marked with schools, to denote locations from which roaming walkers will loop the block during three out of the four legs in their quadrambles. Certainly, a real housing block does not need five water supplies or five schools, so most of those locations could be replaced by housing. The police station can also be replaced by a shrine, once the courthouse is added.

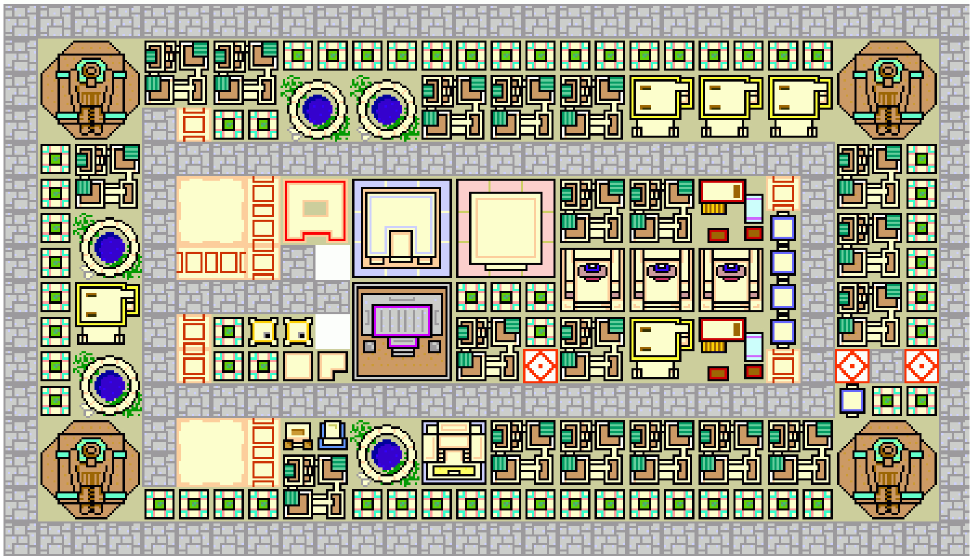

Figure 8. Housing block with synchronized, one-way firemen.

General design elements. The housing block in Fig. 8 has two vital features that allow us to exert a great deal of control over the service-providing roamers in the block. First, the block contains only a single (road-blocked) connection to the external road network (not shown). Second, the block is entirely surrounded by a continuous shell-like road connected through the roadblocks at the entrance to the inner loop. As we will see below, these two features ensure that a great many of the quadramble legs of the roaming walkers in the block are shorted due to blockage and begin with the walker proceeding by the shortest possible route toward the entrance.

One-way firemen, non-confusable long walkers. The firemen from both firehouses in this block always travel around the inner loop in the counterclockwise direction. Therefore, I call them one-way firemen. This unidirectional habit gives better fire suppression than if they always looped but changed directions. The perspective of residents of the house just below the large statue in the upper left corner in Fig. 8 helps to illustrate the advantages of one-way firemen. If a fireman came out of the firehouse alley and turned right to begin a clockwise loop of the block, he would very soon pass the house under the large statue, but could not possibly pass it again until he had traveled more than 50 squares (back to his firehouse and out of the alley again). However, if the fireman turned left the next time he emerged from the alley to begin a counterclockwise loop of the housing block, the house under the statue would have to wait almost 50 more squares of travel before the fireman finally passed it. Over a hundred squares of travel (plus a coffee break for the fireman) is a long time to ask a house to wait without burning at very hard difficulty. If our fireman always looped the block in the same direction, no structure would ever have to wait that long.

Fig. 9 shows the firehouse on the right in Fig. 8 and the roads of the block. Fig. 9 also shows map squares relevant to predicting the walks of the fireman emerging from this firehouse. The four routing centers of the firehouse are marked by small statues; the NW routing center falls in a road square. Around the SW firehouse (bottom of Fig. 9), the algorithm examines the entire innermost search ring, marked by desert squares (including the square under the small statue), without finding a road. In the second search ring, the algorithm finds a road square (green road square) connected to the firehouse so the search stops, and the green road square becomes the target for the SW leg of the quadramble. The shortest circuit from this target to the firehouse has a roadblock on it 28 squares from the firehouse's start square, so the SW leg of the fireman's quadramble shorts due to blockage. The distance from the start square to the target is also greater than 42 squares, so the SW leg of the fireman's quadramble is actually doubly shorted. Fortunately, firemen have never been reported to go on confused walks. Above the NW routing center (left side of Fig. 9), the algorithm finds a connected road square in the second square it searches: the NW walk target shown as another green road square. Again, there is a roadblock along

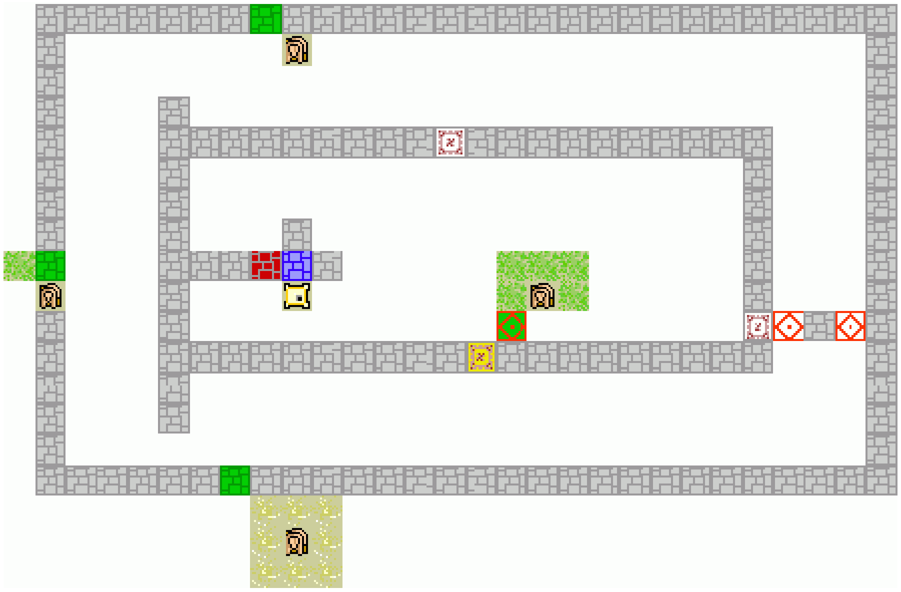

Figure. 9. A one-way fire house and its walk targets.

the short circuit within 43 squares of the firehouse so the NW leg of the fireman's quadramble also shorts due to blockage. Above the NE routing center (top of Fig. 9), the algorithm finds a NE target in the first square it searches. The same roadblocks at the housing block entrance again cause the NE leg of the quadramble to short due to blockage. In the seventh square that the algorithm searches around the SE routing center, the algorithm finds the road square under the green roadblock. If ever a target was blocked, this one is. Thus, all four legs of the firehouse's quadramble are shorted due to blockage.

The SW, NW, and NE legs of the fireman's quadramble function identically. In all three cases, the fireman leaves his (blue) walk start square in destination mode along the 27-square-long shortest road (the lower long road of the main loop in Fig. 9) to the pink plaza square next to the block's entrance. In comparison, it would take him 33 squares to reach the pink plaza if he turned right when emerging from the firehouse alley to take the upper long road to reach the same plaza square. On the pink plaza square next to the block entrance, the fireman shifts to random mode for the remainder of the 43 squares of his roaming walk. This takes him to the pink plaza square on the upper road of the main loop. On this plaza, he reverts back to destination mode with his (red) walk-finish square as his destination. On these three legs of the fireman's quadramble, he is in destination mode while passing through the three-way intersections housing the two pavilions and the bandstand, so he always turns the way we want him to. Over decades of watching this block, I have never seen a fireman enter either stub road.

On the SE leg of the fireman's quadramble, he begins in destination mode but only until he arrives at the (unmarked) road square directly below the green roadblock (one square to the right of the yellow plaza). This is far enough to get him through the band stand and the lower pavilion. Below the green road block, he converts to random mode for the remainder of the 43-square roaming half of his walk, which ends with the fireman on the pink plaza square at the top of the main loop. Again, he converts to destination mode for the return trip to his (red) walk finish square, which carries him safely through the upper pavilion.

Although it is unaesthetic, the green roadblock is essential to ensuring that all four legs of the fireman's quadramble have the same duration. If we removed the green roadblock and the road square beneath it, the algorithm would find the yellow plaza square when it searched for a walk target around the SE routing center. Since there is no roadblock between the firehouse and the yellow plaza, the algorithm would not short out the SE leg of the fireman's quadramble. Instead, a grounded walk would be generated. On this leg, the fireman would travel via the shortest route to the yellow plaza where he would convert to random mode for an additional 43 squares of travel. This brings him to the upper three-way intersection with eight more squares of travel in random mode remaining. If he happened to turn left at this intersection, and then left again at the next (towards his firehouse) he would be very close to his walk-finish square. Or, he could turn right into the upper pavilion's stub road, turn around and turn back along the upper long road of the main loop, in this case he would be much further than a square or two from his walk-finish square when the roaming half of his walk ended. Thus, the duration of the fireman's SE walk could not be relied upon to be the same as the durations of the other three legs in his quadramble. Adding the green roadblock converts the grounded SE leg to a blockage-shorted leg.

The other firehouse is one square to the left of the one shown in Fig. 9. The routing centers of the second firehouse would lie one square to the left of those shown in Fig. 9. The NW and SE walk targets are exactly the same square as they were for the first firehouse. The NE and SW walk targets and one square to the left of the green road squares at the top and bottom of Fig. 9 marking the analogous targets for the first firehouse. Consequently, all four legs of the second firehouse's quadramble short out due to blockage, just as happened for the first firehouse.

Because the second firehouse is one road square closer to the main loop than the first firehouse, the second fireman's total travel distance to complete one leg of his quadramble (57 squares) is actually two squares shorter than the first fireman's 59-square total travel distance. (One square less to reach the nearest three-way intersection from his walk start square, and one square less to reach his walk finish square from the same intersection when returning) One might think that the firemen from these two firehouses could not be synchronized because they travel paths of different length, but this is not true. For firemen, paths of 57 and 59 squares are in the same synchronization group (as described above in the section on Synchronized Firemen, p. 12). If you place the two firehouses at the same time, the two firemen will emerge side by side simultaneously from the firehouses, and then the first firehouse's fireman will follow the second fireman all the way around the block. When they return (second fireman still leading) the second fireman will reach his walk finish square and vanish while the first fireman walks two more squares before vanishing. The second fireman takes a longer coffee break than the first fireman, and shortly they appear simultaneously side by side to repeat the process all over again.

Of course, two fireman are little better than one, if one follows hard on the heels of the other. Much more effective fire danger suppression is achieved if the two fireman are on nearly opposite sides of the block from one another. If the first firehouse is placed first, but the second firehouse is not installed until the first fireman reaches the south corner of the main loop (close to the block's entrance), then the second fireman should follow the first around the loop separated by a little more than 26 road squares. The two fireman will stay separated by this gap for many game decades. I have more than a game century of experience with housing blocks at very hard difficulty with pairs of these completely shorted out, one-way, half-a-leg-out-of-phase firehouses, and have never had fire break out in one of them. Indeed, I ran the block shown in Fig. 8 (with other buildings replacing some water supplies but with all the schools shown) plus the most flammable housing available (cottages) for several decades at very hard difficulty to make sure the fire suppression was rock solid: no fires.

The physician, a non-confusable short walker. Fig. 10 shows the critical squares used in analyzing the quadramble of the physician. As in Fig. 9, the routing centers are marked with small statues, the four walk targets near these routing centers are highlighted as green road squares. Empty search rings are shown as desert squares, and previously examined squares in the same search ring as a target are shown as grass.

The NW and NE legs of the physicians quadramble clearly short out due to blockage, since the shortest circuit to either of these targets must pass through the roadblocks at the block's entrance. Just like the firemen did on their three walks that shorted out on the shell, on his NW and NE legs, the physician heads via the shortest route to the pink plaza square next to the entrance. On that plaza, he converts to random mode and proceeds to the next pink plaza square, at which point he is 26 squares (= *d* for a physician) from his walk start. There, the roaming portion of his walk ends, and he converts to destination mode for the return to his walk finish (red) square. The route to the red square is 25 squares long if he continues around the block in the clockwise direction. If he reversed course and retraced his steps, he would have to travel 27 squares to reach

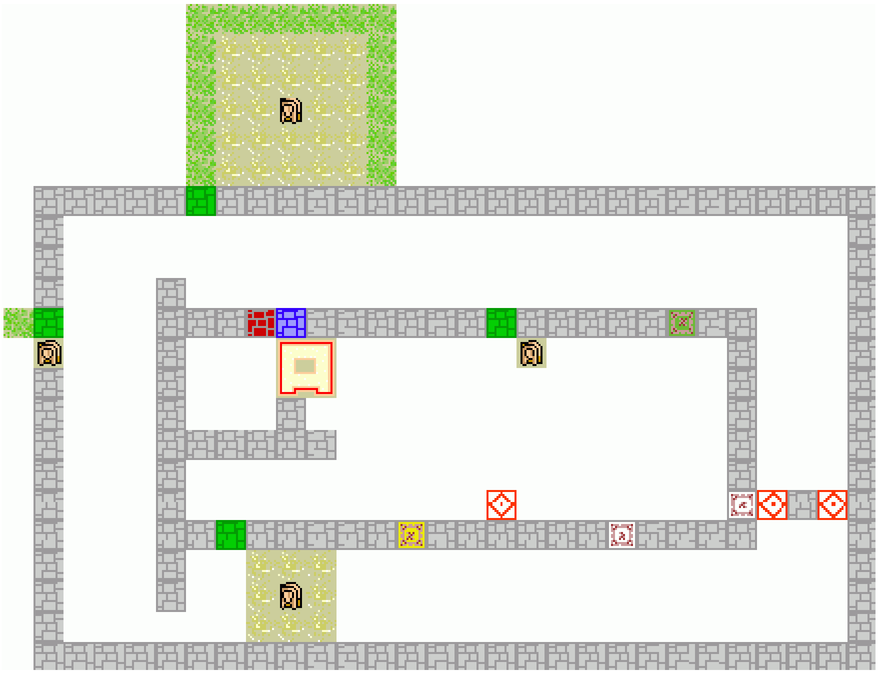

Figure 10. A physician’s office and walk targets. North to upper left.

his walk finish, so he completes the clockwise loop.

The algorithm finds targets on the main loop (green road squares) for the physician’s SE and SW walks. Neither of these targets are blocked so the walks are grounded. On his SE leg, the physician travels in destination mode from his (blue) walk start square to the green road square (towards the right in Fig. 10). On that target he converts to random mode and continues for another 26 squares to the yellow plaza square. He then switches back to destination mode for the trip home to his walk finish. The shortest route home takes him through both pavilions and the bandstand without entering the road stubs.

On the SW leg of the physician’s quadramble he loops the block in the counterclockwise direction. Clearly, the shortest route to his (green) SW walk target runs through all the three-way intersections, so this is how he begins his trip (safely in destination mode while passing through the venues). On his SW walk target he enters random mode for an additional 26 squares of travel to the green plaza square, on which he returns to destination mode for the (short) trip to his walk finish square. Thus, he loops the block clockwise on three of his quadramble legs. Only on his SW leg does he loop the block counterclockwise.

The architect, a confusable long walker. Architects are confusable long walkers. We can count on them remaining visible and effective along the entire length of grounded walks and singly shorted walks, but they may vanish (for quick reuse) or go off road on a doubly shorted (confused) walk after 43 squares of predictable travel. The housing block in Fig. 8, therefore, is arranged to ensure that all damagable buildings and shrines will be passed by the architect within the first 43 squares of his travel on doubly shorted walks. His grounded walks execute complete circuits of the main loop, so they service all the damagable structures as well.

Fig. 11 shows the critical squares for predicting the architect’s behavior. The SW and NW walk targets are clearly shorted due to distance. The SW walk target also lies 49 road squares (via the shortest route) from the architect’s walk start square and the NW target is even further away, so both of these walks are doubly shorted and generate confusion after the walker has traveled his first 43 squares. For both the SW and NW walk, the architect travels via the shortest possible route to the pink plaza square (Fig. 11) next to the block’s entrance and then converts to random mode for the rest of the 43 squares of his predictable outward-bound travel. He reaches the upper three-way intersection with two squares of predictable travel remaining. In my lab, he sometimes turns into the stub then comes back to the intersection and then vanishes. If I remove the

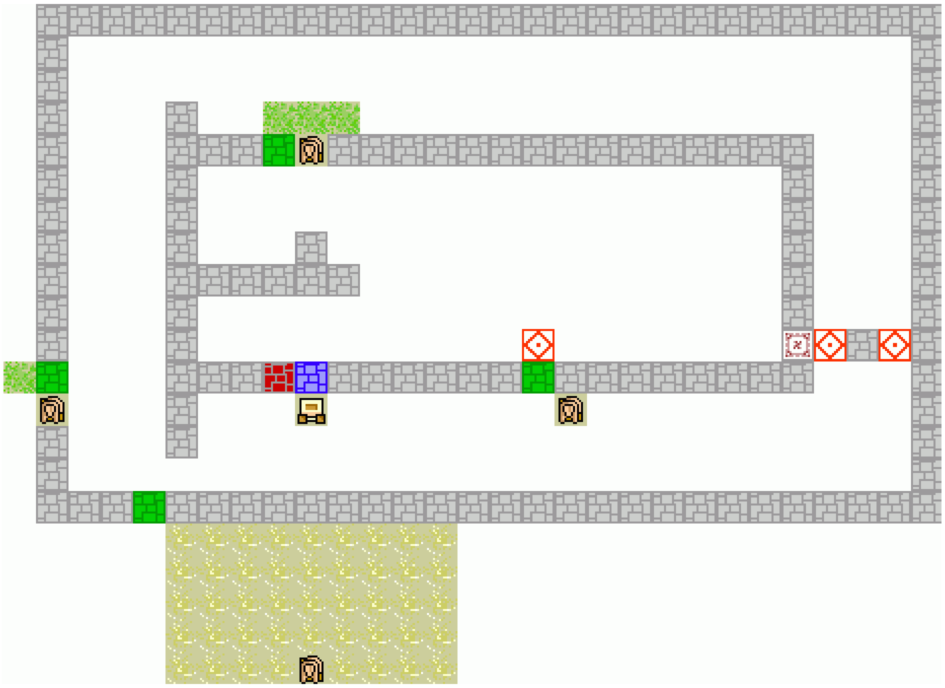

Figure. 11. The architect's office and walk targets.

house under the large statue next to the upper three-way intersecion, the architect frequently goes off road across the squares occupied by the house. Sometimes he turns left, then walks two squares and vanishes. In any event, his behavior after reaching the upper three-way intersection is irrelevant, since he has already passed the courthouse, all the shrines and both temples.

The target for the NE leg of the architect's quadramble (uppermost green road square in Fig. 11) is not at all blocked, so it is used to generate a grounded walk. On this walk, the architect smoothly passes through all of the three-way intersections (without getting distracted by the stubs, since he's in destination mode until he reaches his target) and then converts to random mode for another 43 squares of travel. That is a more than sufficient distance to carry him all the way around the main loop, past all the damagable structures, past his own walk-finish square, and into the lower three-way intersection (Fig. 11) with a few squares of walking left in him. We do not care how he spends these, because he soon returns to his walk-finish square.

The architect also executes a grounded walk for his SE leg. Continuing on for 43 squares beyond his SE target (green road square on the right, Fig. 11) brings him into the uppermost three-way intersection with nine squares remaining to travel, and my architect surely did mess around in the stubs and the firehouse alley before converting to destination mode for the trip back to his walk-finish square. Once again, we don't care if he gets distracted like this, because he has already passed all the damagable structures.

Only the architect's NE walk is guaranteed to pass through all three venue-supporting intersections. The two confusable walks (SW and NW) and the grounded walk to the SE all hit the uppermost intersection (Fig. 11) in random mode, so the architect might reverse course and retrace his steps along the upper long road of the main loop. Therefore, I advise against rearranging the block in Fig. 8 in a way that replaces the school and two water supplies on the left with damagable structures.

The magistrate, a non-confusable long walker. I think that magistrates look marvelously dignified while making their rounds. It would be a shame if the magistrate who patrols the block shown in Fig. 8, embarrassed himself by rattling around in venue stubs. Fig. 12 shows the important squares for the magistrate. The walk targets strikingly resemble those for the (nearby) fireman examined above. The SW, NW, and NE walk targets all lie on the shell road causing those legs of the magistrates quadramble to short due to blockage. Just like the fireman, the magistrate's SE walk shorts out on the

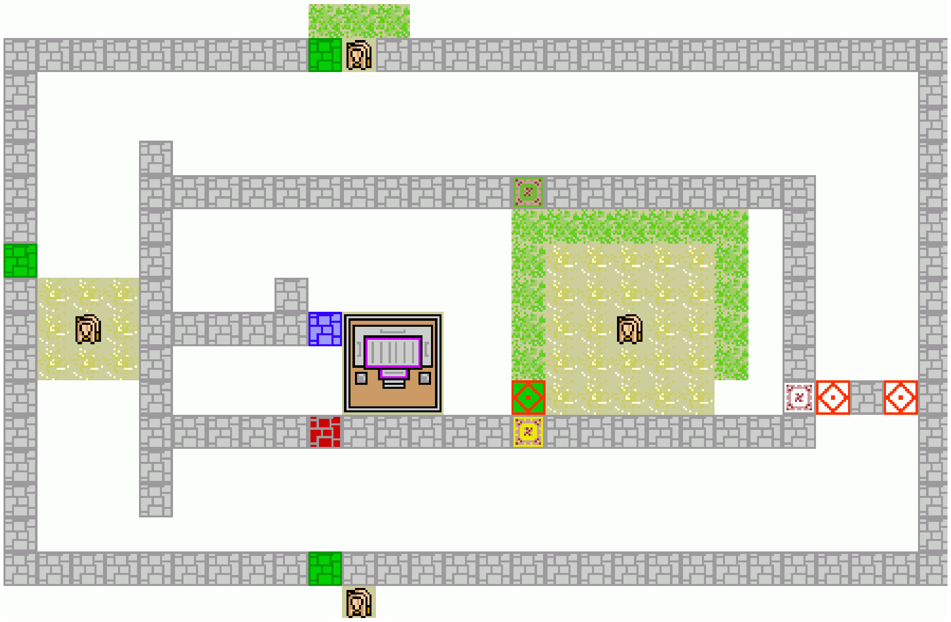

Figure 12. The courthouse and its critical squares.

green roadblock. The SW, NW, and NE walks begin on the blue square and travel (via the lower main loop road) to the pink plaza tile, where conversion to random mode occurs which lasts until the magistrate reaches the green plaza square. On the green plaza, the magistrate converts to destination mode for the return trip to his walk finish (blue road square, again). On the SE leg of his quadramble, the magistrate only proceeds as far as the yellow plaza square in destination mode, but that is still far enough to carry him past two three-way intersections. Just like the other thee legs, the random portion of his SE walk ends on the green plaza tile (43 squares of total travel from his walk start), and he returns to his blue walk finish square in destination mode. At no time does the magistrate enter a three-way intersection in random mode, so he never enters the stub roads.

If we removed the blue road square, the magistrate would use the red road square as his walk start. All four legs of his quadramble would still be blockage shorted, but the 43-square roaming portion of these walks would begin on the red square. He would reach the upper three-way intersection in random mode after 40 total squares of travel from his walk start, and would be able to enter the upper stub road. Since I regarded that as unaesthetic, I prevented it.

Musicians and dancers, confusable medium walkers. The lower pavilion provides better entertainment coverage to the block in Fig. 8 than the upper one does, so this section explores walks and teleportation from the lower pavilion. Fig. 13 shows the routing centers (small statues) and walk targets (and potential teleporting start squares). In the absence of teleportation, any entertainer who is sent on the pavilion's SE quadramble leg executes a grounded walk with the yellow plaza square as the walk target where destination mode converts to random mode for an additional 35 squares of travel to the go-home point one square short of the NE routing center. The entertainer stays visible during his/her destination mode return, since this was a grounded walk. The result is that the entertainer loops the block counterclockwise.

A non-teleporting entertainer dispatched on the NE leg of the pavilion's quadramble executes a grounded walk traveling (in destination mode) from the blue walk start square (Fig. 13) to the green plaza square (the upper pavilion's stub road), where conversion to random mode for 35-more squares of travel occurs. The entertainer cannot continue in the same direction he had been going because he has run out of road and must return, instead, to the intersection. This is pretty much a wasted walk, since we cannot predict where the entertainer is going to go in random mode at this intersection.

A non-teleporting entertainer executing either the SW or NW leg of the pavilion's quadramble will

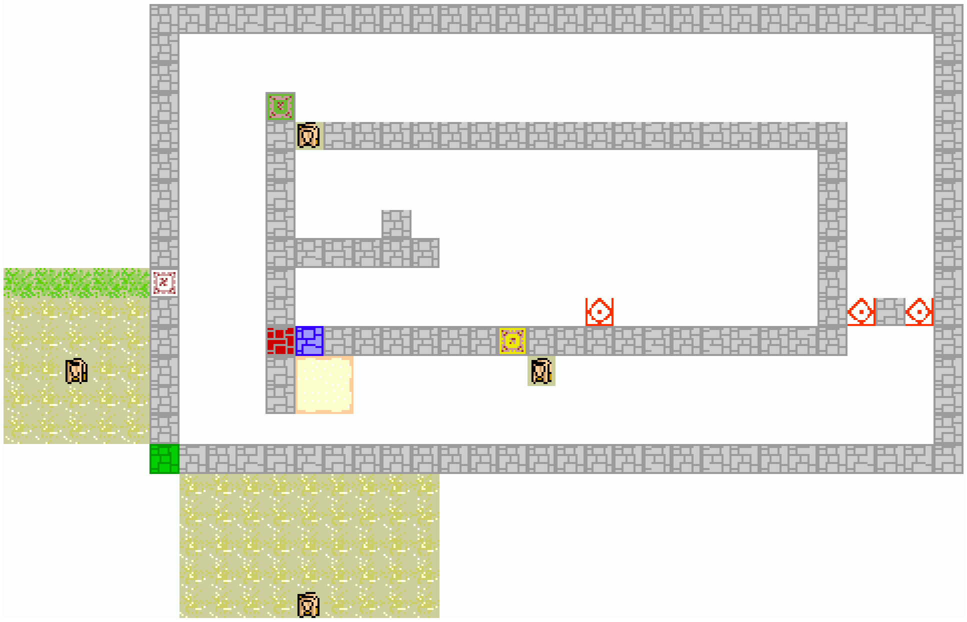

Figure 13. Critical squares for a pavilion with the dance stage located as shown. Not all of the squares searched around the SW search center (small statue) that are examined before the SW walk target is discovered by the algorithm are shown.

be doubly shorted (blockage and distance) and will enter confused mode after 35 squares of travel. The walk begins in destination mode to the main loop road square next to the roadblock at the block's entrance. The entertainer continues in random mode until a total of 35 squares have been walked from the blue walk start square. These 35 squares of travel carry the entertainer a little beyond the midpoint along the upper long road of the main loop. At that point, the entertainer enters confused mode, and his or her behavior cannot be predicted with confidence. When I built this block and ran it for several decades, I observed all three types of pavilion-generated entertainers frequently executing this walk. In practice, the entertain would continue straight along the upper long road of the main loop and vanish right on the three-way intersection of the upper pavilion, presumably because her or she ran into the residence at the end of the street. In the absence of that house, I would expect the entertainer to continue going straight, right off the end of the road.

Teleportation from the Pavilion. Although the yellow and green plaza squares were certainly used as teleported walk-start squares in my housing block of the design shown in Fig. 8, the green road square (in the lower left hand corner of Fig. 13) and the pink plaza square above it were not. This was because my city had industrial block roads in the areas shown as desert squares in Fig. 13, so the algorithm found higher priority road squares that became the actual walk targets or teleporting walk-start squares on these two walks. A dancer or musician who uses the green plaza square in the upper stub road of Fig. 13 as a teleporting walk-start square will make a grounded walk with the yellow plaza square as the walk target that will enable the entertainer to loop the block in the counterclockwise direction.

A dancer or musician who uses the yellow plaza square as a teleporting walk-start square will have a walk target lying somewhere in the lower region of desert squares below the dance stage in Fig. 13. The walk will be confused by double shorting (due to distance and to the roadblocks at the entrance to the housing block). The entertainer will travel (in destination mode) from the blue walk-start square to the inner loop road square right next to the roadblock at the entrance. On this square the entertainer switches to random mode and continues up the road, around the corner, and NW along the upper long road of the inner loop. At little after the midpoint along this road (after they walker has traveled 35 squares from the walk start), the entertainer switches to confused mode and becomes unpredictable. In the actual block that I ran, the confused entertainer simply continued walking in the same direction until he or she arrived at the three-way intersection of the upper pavilion and then vanished.

If a musician or dancer used the green road square (lower left of Fig. 13) as a teleporting walk-start square, then he or she would execute a grounded walk with the pink plaza square as a walk target. In practice, the odds are good that there will be roadblocks between the SW target and the NW target squares, so this walk will probably short out due to blockage, and in my city it shorted doubly and eventually sent the entertainers who executed this walk out into the weeds in confusion. This is not a useful walk. Walks using the pink plaza tile as a teleporting walk start square are similarly useless. Such an entertainer would have the green plaza tile as a walk target In the absence of roadblocks on the shell road, the entertainer would travel 35 squares along the shortest route (lower long shell road) heading towards the roadblocks at the entrance to the housing block before becoming confused. In actual practice, my entertainers never made it to the housing block's entrance because they quickly ran into roadblocks.

The combination of terrestrial and teleporting walks that issued from a single pavilion with a dance stage located as shown in Fig. 13 kept my housing block (of the design shown in Fig. 8) stable for decades. Certainly the houses adjoining the lower long main loop road were kept saturated for entertainment by all the newly arrived entertainers walking from the entrance to the pavilion. If a player felt uncomfortable relying on the lower pavilion alone (with all the inherent unpredictability associated with teleportation), installing the upper pavilion will guarantee that newly arriving entertainers will also pass by all the houses along the upper long road of the main loop.

In addition to running the housing block shown in Fig. 8 for decades in one of my walker labs, I also recently ran a slight modification of it in Sauty (again for decades, because I am slow) built at very hard difficulty. The firehouses worked perfectly and never needed to be reset. The lower pavilion alone was enough to keep a full load of elegant residences entertained and stable. Towards the end of the game when I needed to boost my culture rating, the presence of the other three-way intersections as sites for another pavilion and a bandstand proved invaluable.